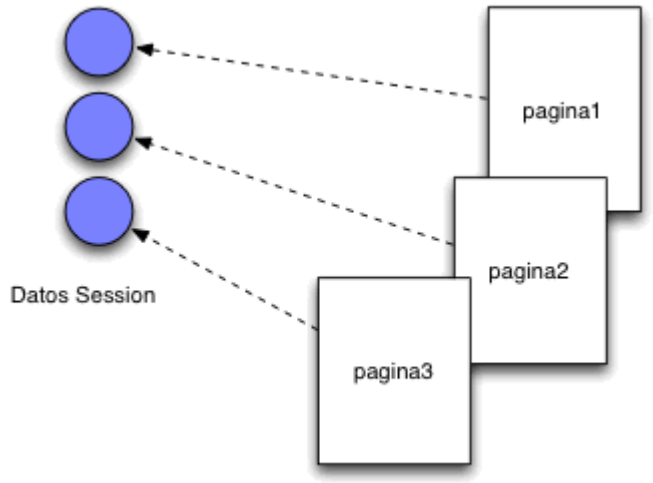
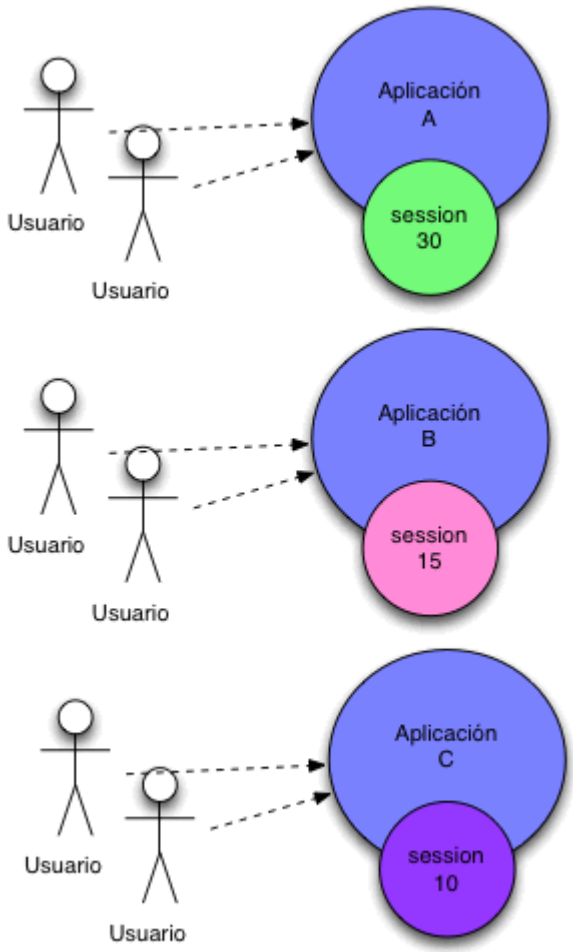
El objeto de Java HttpSession es uno de los más utilizados en aplicaciones web ya que continuamente necesitamos almacenar información que este viva para un usuario concreto mientras este maneja la aplicación y se mueve a través de las distintas páginas.



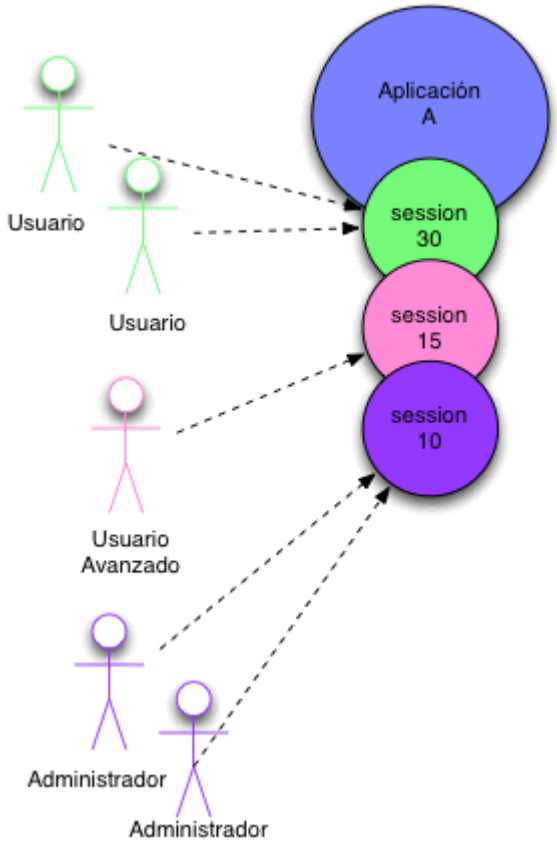
En estas situaciones nos encontramos que es clave saber cuanto tiempo esta viva una session determinada. Esto se configura normalmente a través del web.xml con las siguientes etiquetas.

```
<session-config>
 <session-timeout>30</session-timeout>
</session-config>
```

El Tomcat tiene configurado por defecto un timeout de 30 minutos. Sin embargo en muchas ocasiones queremos modificarlo ya sea para bajarlo y reducir el timeout como para subirlo,dependerá mucho del tipo de aplicaciones.



Ahora bien este tipo de configuración aunque es muy sencilla tiene una serie de limitaciones la más importante es que trata a todas las sessiones de todos los usuarios por igual. Quizás esto no sea lo que deseemos ya que hay situaciones que la sesión de un usuario normal es razonable que dure 30 minutos pero la session de un administrador debe durar solo 5. ¿Como abordamos este problema?.



Para personalizar el tiempo de timeout de un usuario concreto podemos atacar directamente al objeto HttpSession y a su método setMaxInactiveInverval(int segundos) que nos permite configurar el tiempo de timeout de forma individual para cada usuario (en segundos) vamos a verlo en código:

```
package com.arquitecturajava;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/Servlet1")
public class Servlet1 extends HttpServlet {
 private static final long serialVersionUID = 1L;

 protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

 HttpSession session=request.getSession();
 session.setMaxInactiveInterval(60);
 session.setAttribute("nombre", "pedro");

 }

}
```

Este primer servlet almacena un objeto de tipo String (“nombre”) en la session pero modifica su tiempo de timeout para que solo dure 60 segundos. Usaremos otro servlet para leer la información.

```
package com.arquitecturajava;

import java.io.IOException;

import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/Servlet1")
public class Servlet1 extends HttpServlet {
 private static final long serialVersionUID = 1L;

 /**
 * @see HttpServlet#HttpServlet()
 */
 public Servlet1() {
 super();
 // TODO Auto-generated constructor stub
 }

 /**
 * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
 */
 protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

 HttpSession session=request.getSession();

 String nombre=session.getAttribute("nombre", "pedro");
PrinWriter pw=response.getWriter();
pw.println(nombre);
 }
```

```
}
```

Este segundo servlet nos imprimirá los datos almacenados en la session por pantalla pero en vez de durar la session 30 minutos únicamente durará 60 segundos (un minuto) una vez pasado el minuto los datos se perderán.

De esta forma tendremos un control mucho más a detalle de la gestión de sesiones.

Otros artículos relacionados : El concepto de HttpSession , Manejando ServletContextListener